BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

INTERRUPTION CONTROL SYSTEM FOR PROGRAM MODULE

Patent Number: JP4235642
Publication date: 1992-08-24
Inventor(s): YAMADA MASANORI; others: 01
Applicant(s):: HITACHI LTD
Requested Patent: JP4235642
Application Number: JP19910001421 19910110
Priority Number(s):
IPC Classification: G06F11/28
EC Classification:
Equivalents:

Abstract

PURPOSE:To interrupt the execution of a program with a called module when a program module calls another program.
CONSTITUTION:The position of an interruption point is set through 8 call graph control part 301 where an interruption module is set with an instruction of a user and a screen input/output part 302 where the display end the input are carried out on a display screen 306. An interruption control part 303 sets an interruption point to a debug subject program 304 based on the position of the interruption point set at the part 301. Meanwhile the program 304 discriminates whether the coincidence is secured or not between the module called in an interruption state and a call module received an interruption request from the user. Then a program module can be interrupted with a combination of call modules and in such a constitution including the part 303, the part 302, end the debug information 305.

TOP

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号

特開平4－235642

(43)公開日　平成４年(1992) ８月24日

(51)Int.Cl.⁵	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　11/28	３１５　Ａ	7165－5Ｂ		

審査請求　未請求　請求項の数２(全　４　頁)

(21)出願番号　特願平3－1421

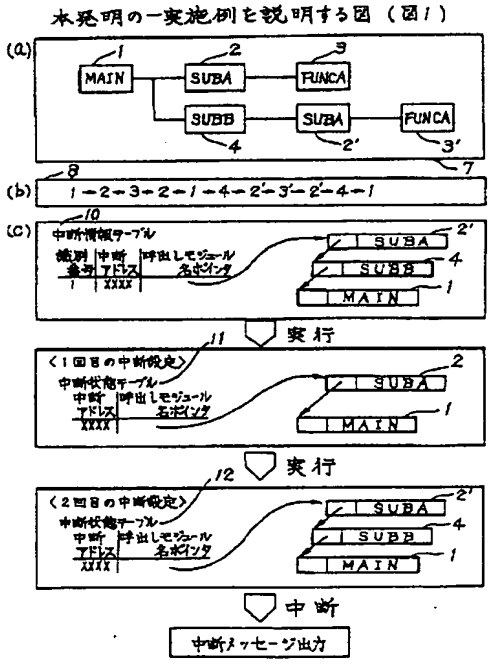
(22)出願日　平成３年(1991) １月10日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　山田　雅則
神奈川県横浜市戸塚区戸塚町5030番地株式会社日立製作所ソフトウエア工場内
(72)発明者　磧谷　利夫
神奈川県横浜市戸塚区戸塚町5030番地株式会社日立製作所ソフトウエア工場内
(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　プログラム・モジュールの中断制御方式

(57)【要約】
【目的】本発明はプログラム・モジュールが別のプログラム・モジュールを呼出すとき、呼出したモジュールによってプログラム実行を中断させることにある。
【構成】中断指示をしたプログラム・モジュールに制御を移す時点でこのプログラム・モジュールを呼出した上位のプログラム・モジュールがあらかじめ定められたものであれば、実行中断をすることにより達成される。
【効果】不良発生の原因となるようなモジュール呼出し状態でプログラム実行を中断することができる。

本発明の一実施例を説明する図（図１）



1

【特許請求の範囲】

【請求項1】実行中のプログラムが指定された中断点に達したときプログラム実行を中断する中断制御方式において、中断指示をしたプログラム・モジュールに制御を移す時点で前記プログラム・モジュールを呼出した上位のプログラム・モジュールがあらかじめ定められたものであるかどうかを判定し、それによって前記プログラム・モジュールを中断するか否かを決定することを特徴とするプログラム・モジュールの中断制御方式。

【請求項2】前記上位のプログラム・モジュールは複数個あり、全体として呼出し関係のチェインを構成していることを特徴とする請求項1記載の中断制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、プログラムデバッガのモジュールを単位としたデバッグ方法における中断制御方式に関する。

【0002】

【従来の技術】従来の技術は、「プログラムプロダクトVOS3　FORTRAN77テストデバッグ支援FORT77／TD解説／手引書」（（株）日立製作所昭和62年4月発行）の第30頁～第32頁に記載のように、プログラムの中断処理は、中断したい位置を実行した時、無条件に中断していた。従ってあるプログラム・モジュールが別のプログラム・モジュールを呼出すとき、呼出される別のモジュールの開始位置で中断することはできるが、呼出したモジュールが何かによって中断することはできなかった。

【0003】

【発明が解決しようとする課題】上記従来技術は、モジュールを呼出した状態によって不良となったり、ならなかったりするような種類の不良に対して、単に中断位置を指定することで中断していたため、呼出したモジュールが配慮されておらず、不良発生の原因となる状態で中断されることができなかった。

【0004】本発明の目的は、プログラム・モジュールが別のプログラム・モジュールを呼出すとき、呼出したモジュールによってプログラム実行を中断させることにある。

【0005】

【課題を解決するための手段】本発明は、中断指示をしたプログラム・モジュールに制御を移す時点でこのプログラム・モジュールを呼出した上位のプログラム・モジュールがあらかじめ定められたものであるかどうかを判定し、あらかじめ定められたものであるとき実際の中断を行うプログラム・モジュールの中断制御方式を特徴とする。

【0006】

【作用】中断指示をしたモジュールは、上位のモジュールによって複数回呼出されて実行される可能性がある。

2

目的のモジュールの中断指示をするだけでなく、このモジュールを呼出した上位のモジュールをあらかじめ設定しておき、目的のモジュールに制御を移す時点でこの呼出しモジュールがあらかじめ定めたものであるかどうかチェックし、あらかじめ定めたものであるとき中断することにより、不良発生の原因となるようなモジュール呼出し状態でプログラム実行を中断することができる。

【0007】

【実施例】以下、本発明の一実施例を詳細に説明する。

【0008】図1は、本発明をもちいた一実施例であり、図2は、中断にいたるまでの処理の流れを示すフローチャートであり、図3は、中断制御をするデバッガの構成図である。

【0009】図3において、利用者からの指示により中断モジュールを設定するためのコールグラフ制御部301とそれをディスプレイ画面306に表示や入力をするための画面入出力部302により、中断点の位置を設定する。ここでコールグラフとは、モジュールの呼出し関係を示す情報である。中断制御部303は、コールグラフ制御部301により設定された中断点位置を元にデバッグ対象プログラム304に中断点を設定する。また、デバッグ対象プログラム304で中断した時の呼出しモジュールと利用者から中断要求のあった呼出しモジュールが一致するか否かにより、中断するか否かを判定する。中断制御部303、コールグラフ制御部301、画面入出力部302及びデバッグ情報305の構成によって呼出しモジュールの組合せによる中断が実現できる。

【0010】図1は、図2に示す中断にいたるまでのクローチャートの手順にしたがって行った一実施例を示している。例とするプログラムは、モジュールMAIN，SUBA，SUBB，FUNCAから成り、モジュール構成7のようにリンクされており、実行順序8に示す順序で実行する。まず、MAINモジュール1からSUBBモジュール4，SUBAモジュール2′の順に呼出された時にFUNCAモジュール3′で不良が発生すると仮定すると、中断点情報の設定としてFUNCAモジュール3′に制御を移す時点で中断するようにする。この時に、識別番号と中断アドレスと呼出しモジュール名ポインタを登録した中断情報テーブル10を作成する。図に示すように、上位の呼出しモジュール名はポインタによって順にチェインにつながれている。

【0011】次にデバッグ対象プログラム304の実行を行ない中断アドレスによって中断するのを待つ。図1の実施例では、まずFUNCAモジュール3の開始位置で一時的な中断をする。このとき、中断時の呼出しモジュール名を取出し、デバッグ対象プログラムの中断状態テーブル11を作成する。呼出しモジュール名はポインタによってチェインにつながれている。

【0012】次に中断をするか否かを判定するために、中断状態テーブル11の中断アドレスと中断情報テーブ

3

ル１０の中断アドレスと一致するものを検索する。それから、一致した中断情報テーブル１０中の一連の呼出しモジュール名が中断状態テーブル１１中の一連の呼出しモジュール名と一致するか判定する。１回目の中断判定では、中断状態テーブル１１の呼出しモジュール名のチェインと中断情報テーブル１０の呼出しモジュール名のチェインが不一致であるため、一時的な中断状態を解除してデバッグ対象プログラムの実行を再開させる。ＦＵＮＣＡモジュール３′を実行するとき、中断アドレスによって再び一時的な中断をするので、このとき中断状態テーブル１２を作成する。呼出しモジュール名はポインタによって順にチェインにつながれている。

【００１３】次に行われる中断判定では、中断状態テーブル１２の呼出しモジュール名のチェインと中断情報テーブル１０の呼出しモジュール名のチェインが一致しているので、デバッグ対象プログラムが本格的な実行中断状態となる。本実施例によれば、中断と実行の操作が１回少なくなる効果がある。一般には、大規模なプログラムであるほど部品化されたプログラムの呼出し箇所が多くなるので、中断と実行の操作回数が、呼出し箇所に比例して減少する。

【００１４】図２は、中断にいたるまでのデバッガの処理を示すフローチャートである。まず、中断点情報の設定２０１によって利用者から指定された位置の中断アドレスと呼出しモジュール名ポインタを格納した中断情報テーブル１０を作成する。

【００１５】次は、プログラムの実行２０２によってデバッグ対象プログラムを実行する。デバッグ対象プログラムが、中断情報テーブル１０の中断アドレスで実行待ちになるまで待つ。

【００１６】中断アドレスによって中断した時に呼出しモジュール名取出し２０３へ処理が移り、中断時の呼出しモジュール名ポインタを取出し、中断アドレスと呼出しモジュール名ポインタを格納する中断状態テーブル１

4

１，１２を作成する。

【００１７】次に、ステップ２０４において、中断設定時に作成した中断情報テーブル１０の呼出しモジュール名チェインと実行中断時に作成した中断状態テーブル１１，１２の呼出しモジュール名のチェインを比較して、一致していれば中断処理２０５へ、一致していなければプログラムの実行２０２に戻ってプログラムの実行を続ける。これにより、デバッガの操作回数を削減することができる。

【００１８】なお呼出しモジュール名の比較は、呼出し関係の近いものから順に上位の方向に行なうことはいうまでもない。

【００１９】

【発明の効果】本発明によれば、呼出しモジュールの呼出し関係によって中断できるので、中断と実行を繰り返す操作の簡略化の効果がある。

【００２０】また、モジュールレベルの中断ができるので、デバッグ作業の効率も向上する効果がある。

【００２１】さらに、モジュール単位のデバッグができるのでプログラムの部品化への効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例を説明する図である。

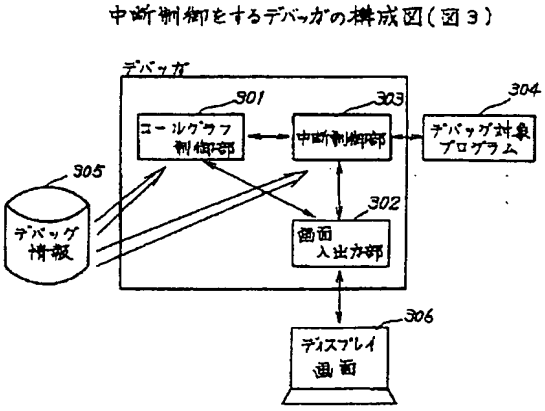
【図２】中断にいたるまでの処理の流れを示すフローチャートである。

【図３】中断制御をするデバッガの構成図である。

【符号の説明】

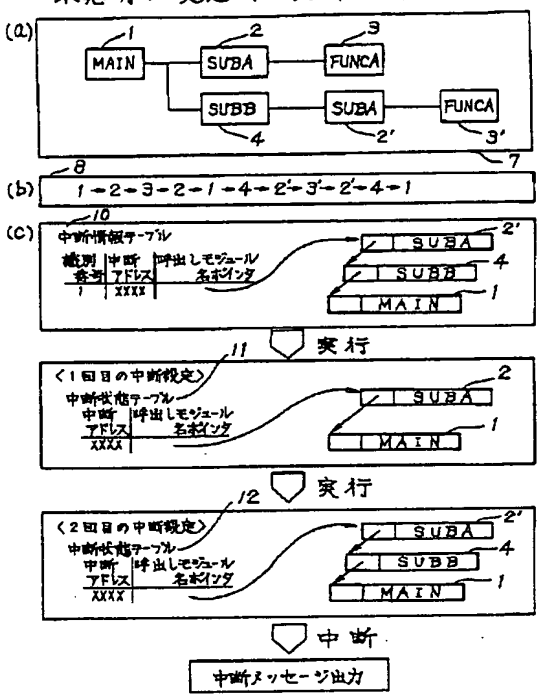
１０．中断情報テーブル
１１．中断状態テーブル
１２．中断状態テーブル
３０１．コールグラフ制御部
３０２．画面入出力部
３０３．中断制御部
３０４．デバッグ対象プログラム

【図３】



【図1】

本発明の一実施例を説明する図（図1）



【図2】

中断にいたるまでのフローチャート（図2）